

CMPE 327 Test Plan

How did your team organize the documentations of the test cases

Main branch contains the following two files related to testing

- A1_Testcases.md
- A1_Testcase_Summary.md

A1_Testcases.md contains all testcases in natural language form following the format given in examples. A1_Testcase_Summary.md is a markdown table that summarizes the created test cases. The markdown table includes a Test Overseer column representing the team member overlooking the defined testcase. Columns in this table include, Specification, ID, Purpose, Pass Criteria, Client Ask, Test Overseer.

Your understanding of how the chosen testing framework works to test the frontend, including your understandings of when and how the test cases will be running directly on GitHub.

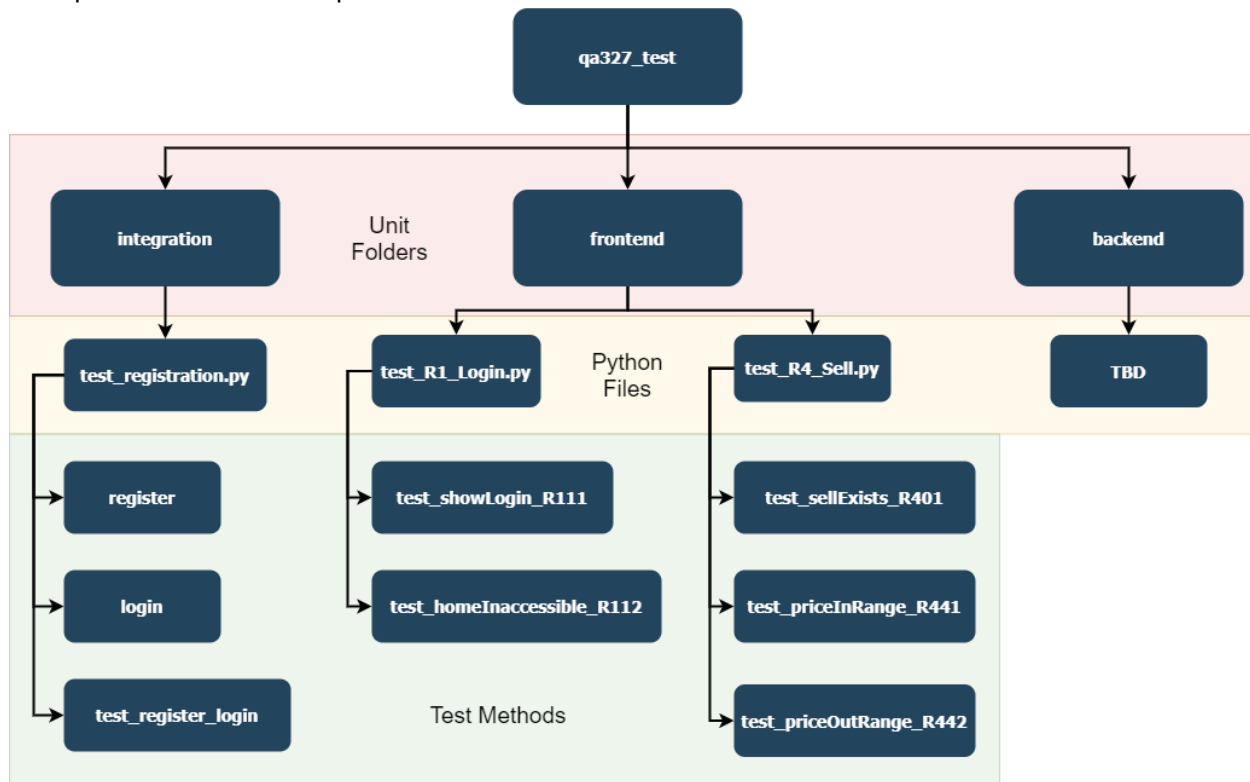
The front end of the framework is tested using pytest as it is responsible for making sure that our code is still able to setup a connection, login and reach the homepage. This is most important as all the other testcases would not be able to run if the server fails to startup or you are not able to login. To test the frontend the conftest.py file is needed as it contains fixtures that are shared between classes and used in the server setup. Within conftest.py a new thread is created to run the web service/server, this thread will run until all test cases have completed. Where it will then close the server, completing the test suite.

Testcases will be run directly from Github using Github Actions. We'll define the workflow of our CI process from the template provided in our repository. With it, we automatically run the testcases on the cloud after changes to the codebase. Test runs will also include logs from the test runs for debugging.

How are you going to organize different test case code files? (a folder for a specification?)

Test scripting will be divided into folders frontend, backend, and integration, with each folder containing test files related to each requirement/constraint list (R1 /login, R2 /register. Etc.) provided by the customer. These three folders will be stored under qa327_test directory. Test case code files will each begin with 'test_' to satisfy the automated test running requirements of Selenium Base. Each test case name will be a condensed version of the test purpose (written in camel case) followed by an underscore and its unique test case numerical identifier. The purpose and expected result for each test case will be included as block comments before the testing code.

Example folder structure is provided here:



Example outline for single test case:

```
@patch('') # Any patch/mocking
def test_nameOfTest_R000(self): # name of test including test_ and numerical
    identifier
    '''Description of purpose of test and expected passing criteria'''
    # Test code goes here
```